

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-076816

(43)Date of publication of application : 14.03.2003

(51)Int.Cl.

G06F 17/60

G06F 9/44

(21)Application number : 2001-264993

(71)Applicant : FUJITSU LTD

(22)Date of filing : 31.08.2001

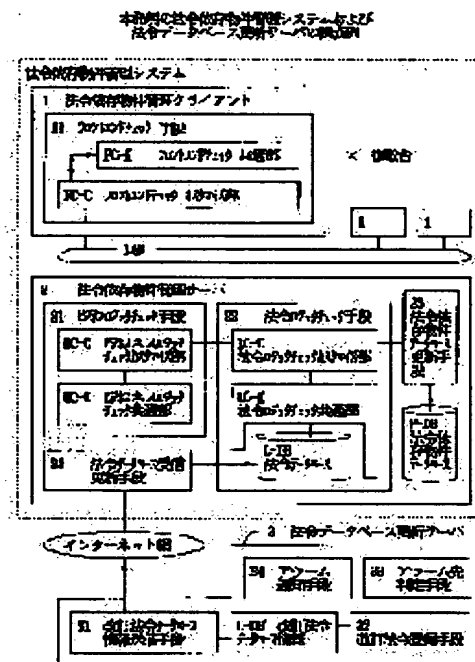
(72)Inventor : IETOMI SHINJI  
ITO KUNIAKI  
OGAWA AKIHITO  
TAKEHARA TAKESHI  
ABE YUTAKA

## (54) STATUTE-DEPENDENT OBJECT MANAGEMENT PROGRAM, STATUTE-DEPENDENT OBJECT MANAGEMENT SYSTEM AND STATUTE DATABASE UPDATE SERVER

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To solve a problem that, for a conventional statute-dependent object management system (for instance, patent management system), a considerable amount of man-hour is required for precise and speedy version upgrade of the system package by a vendor at every statute revision, and a user also necessitates a considerable amount of man-hour for customizing associated with the version upgrade.

**SOLUTION:** A statute-dependent object management program is separated into a business logic check program and a statute logic check program, further, the statute logic check program is separated into a statute logic check common part and a statute logic check customizing part, and a statute database part is separated from the statute logic check program, so that the range of version upgrade at the statute revision can be limited to the statute logic check program, the statute logic check common part and the statute database part, revision misses can be reduced, and the man-hour can be reduced.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

**\* NOTICES \***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

**CLAIMS**

---

[Claim(s)]

[Claim 1] The statute dependence housing management server which manages the procedure which has the statute dependence housing database which stores the tendency hysteresis of a statute dependence housing, and is needed based on a statute, In the statute dependence housing managerial system which consists of one or more statute dependence housing management clients which perform the data registration to said statute dependence housing database, and an updating input It is a statute dependence housing manager for making the computer of a statute dependence housing management server perform statute dependence housing management information processing. As opposed to the data input for the data registration to said statute dependence housing database from said statute dependence housing management client, and updating It checks according to the business logic which is the Ruhr in management business appointed beforehand. The business logic checking program which receives only the data which will stimulate reinput if there is an error, and do not have the error as business logic and as which a computer is operated as a business logic check means, To said data input, if there is an error in the light of the statute logic which is the Ruhr in a statute appointed beforehand, reinput will be urged. The statute logic checking program which receives only data without a statute logic error and as which a computer is operated as a statute logic check means, The statute dependence housing manager characterized by \*\*\*\*(ing) and said business logic checking program and said statute logic checking program having dissociated.

[Claim 2]

## \* NOTICES \*

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2. \*\*\*\* shows the word which can not be translated.

3. In the drawings, any words are not translated.

## DETAILED DESCRIPTION

## [Detailed Description of the Invention]

[0001]

[Field of the Invention] Like the patent managerial system which manages the tendency of a its company application patent in technical open promotion business etc., this invention relates to the construction technique of a statute dependence housing managerial system of managing a statute dependence housing (for example, applied patent), and relates to the technique of performing improvement in a system accompanying revision of a statute efficiently especially.

[0002]

[Description of the Prior Art] It is needed for a global ED competitive age, and the patent strategy has turned into a part of important management strategy in technical open promotion business at it especially. In the technical open promotion business which makes the world a market, patent application is strategically performed to each country of not only Japan but the U.S., Europe, and Asia, and suitable right acquisition is aimed at. Thus, in the company with much application number of cases, for patent strategy promotion, IT technique was used and systematization of patent management business has been attained. drawing 2 (a) \*\*\*\* — the whole aspect of systematization of such patent management business is shown.

[0003] It is used, in case it has are recording of a 1. well-known example retrieval system: CD-ROM official report, and a retrieval function and an artificer and an inner review section examine a their company patent idea as a typical system, as shown in drawing.

2. Artificer application support system : it has functions, such as an in-house application request, personal computer application, and office cooperation, and support application business.

3. Patent managerial system : perform in-house reception of the application request from an artificer, a status control, pension management, cost management, etc., and manage the tendency hysteresis of an application patent.

4. Bag managerial system : it has electronic storage of an application document and a procedure document and a retrieval function, and becomes a tool at the time of a problem patent or patent dispute.

[0004] This invention relates to 3. patent managerial system among the above-mentioned range. A patent managerial system manages the tendency hysteresis in the life cycle of the patent from a patent birth or before to right expiration in a patent database. Although it \*\* in a system at a patent life cycle and much patent management business is included, it is drawing 2 (b). The typical operating list is shown. This invention relates to the patent management business which updates patent database P-DB of a patent managerial system before long, and is drawing 2 (b). Although it is the part which attached O, this is explained later.

[0005] Although these patent managerial systems are introduced by much technical open promotion business etc., those many purchase the patent management system package which a system vendor offers, they customize so that a their company environment, an in-house patent management plan, and the Ruhr may be suited, and are employing this system. However, many differed for each country, and since there are many classes, such as patent right and a service mark right, the statute (it is called a patent statute below) by which industrial rights including patent right are provided has taken in 30 or more countries and 100 sorts or more of patent statutes in the large-scale patent management system user. Of course, a system vendor supports these [ all ].

[0006] Furthermore, reflecting the pro patent age, these statutes mix small revision and large revision together, and are revised frequently. For example, in Japanese Patent Law, it continues in Heisei 5 also in recent years with amendment, the Heisei 6 amendment, the Heisei 10 amendment, and the Heisei 11 amendment. And generally, since patent management requires long-term management, it needs management of the incident patent which applies numbers of generations front Patent Law, so that the present Japan patent period may be 20 years from application.

[0007] In the statute dependence housing managerial system which manages the housing (for example, applied patent) depending on many statutes like the patent managerial system described above If the statute with which a variety is related as mentioned above is held, one - two revision per year will be encountered. In the devotion to the version up activity of a package being required of a system vendor at every the revision of the, and applying a version up system to a user further, the customize part of the user concerned is grasped for every user. A new customize activity was required so that it might have consistency in the version up concerned.

[0008]

[Problem(s) to be Solved by the Invention] Thus, in a statute dependence housing managerial system which is represented by the conventional patent managerial system, the technical problem that a man day great for the

version up of a quick system package with an exact vendor to whenever [ of statute revision ] was required, and the customize accompanying this version up also took a user a great man day occurred. It aims at offering the statute dependence housing managerial system which this invention solves [ managerial system ] such a technical problem, decreases more the version up range of the package accompanying statute revision, and decreases the leakage in revised reflection by reduction and a mistake of an activity man day. Moreover, it aims at offering the statute dependence housing managerial system which makes the activity man day of the user customize accompanying version up of a package reduce.

[0009]

[Means for Solving the Problem] The statute dependence housing management server 2 which manages the procedure which has statute dependence housing database P-DB which stores the tendency hysteresis of a statute dependence housing, and is needed based on a statute as the above-mentioned technical problem is shown in drawing 1 . In the statute dependence housing managerial system which consists of one or more statute dependence housing management clients 1 which perform the data registration to said statute dependence housing database P-DB, and an updating input It is a statute dependence housing manager for making the computer of the statute dependence housing management server 2 perform statute dependence housing management information processing. As opposed to the data input for the data registration to said statute dependence housing database P-DB from said statute dependence housing management client 1, and updating It checks according to the business logic which is the Ruhr in management business appointed beforehand. The business logic checking program which receives only the data which will stimulate reinput if there is an error, and do not have the error as business logic and as which a computer is operated as a business logic check means 21, Reinput will be urged if there is an error in the light of the statute logic which is the Ruhr in a statute appointed beforehand to said data input. The statute logic checking program which receives only data without a statute logic error and as which a computer is operated as a statute logic check means 22, It \*\*\*\* and, therefore, the statute dependence housing manager which said business logic checking program and said statute logic checking program have separated is solved.

[0010] Moreover, business logic check common section BC-K which is the abstract class object set said business logic check function (function of the business logic check means 21) describes business logic check processing to be in common, The program which consisted of business logic check customize section BC-C customized so that it might consist of an activation class for performing said business logic check common section BC-K and the system environment and the operating conditions for every user might be suited realizes. Statute logic check common section LC-K which is the abstract class object set said statute logic check function (function of the statute logic check means 22) describes statute logic check processing to be in common, So that it may consist of an activation class for performing said statute logic check common section LC-K and the system environment and the operating conditions for every user may be suited Therefore, the above-mentioned technical problem is solved by the statute dependence housing manager realized by the program which consisted of customized statute logic check customize section LC-C.

[0011] moreover, the statute dependence housing manager which dissociates with statute logic check common section LC-K, and holds statute database L-DB constituted as a set of an item which was suitable in order that a statute logic check function (function of the statute logic check means 22) might make a computer carry out the logic check of the predetermined description of a statute — the above-mentioned technical problem is solved as be alike.

[0012] Moreover, a revised statute registration means 32 to register as revised statute database information constituted as a set of an item which was suitable in order to make a computer carry out the logic check of the predetermined description of the revised statute, From the statute data-base-updating server 3 which has a revised statute database information transmitting means 31 to transmit said revised statute database information through the Internet to a statute dependence housing managerial system Therefore, the above-mentioned technical problem is solved by the statute dependence housing managerial system which has a renewal means 24 of statute database reception to receive revised statute database information through the Internet, and to update statute database L-DB.

[0013] About the above operation, it \*\* in the example, and explains in full detail below.

[0014]

[Embodiment of the Invention] Drawing 1 - drawing 12 explain the example of the statute dependence housing manager of this invention, a statute dependence housing managerial system, and a statute data-base-updating server. In addition, although computer processing in this invention is performed by the computer program on the main storage of the computer concerned, the offer gestalt of this computer program begins the auxiliary storage unit connected to the computer concerned, is stored in each record medium, such as portable mold storage, such as CD-ROM, and main storage of other computers by which network connection was carried out, an auxiliary storage unit, is offered, and on the occasion of activation of this computer program, on the main storage of the computer concerned, loading of it is carried out and it is performed.

[0015] It is drawing 2 (b) previously. As explained, the patent management business which updates patent database P-DB is various. if the operating outline is extracted and shown below — a open check (the open number of an application patent —) a open day input and request-for-examination decision (the request-for-examination necessity of the application patent by inner review —) The decision result input of a claim stage and amendment existence, a request for examination (a claim day, other office data input). They are \*\*\*\*\* processing (various notice acceptance inputs, such as a notice of the reason for refusal), intermediate processing intermediate

treatment (input of the correspondence action decision result to \*\* by the section for in the company), an amendment procedure (input of the shipping date of a revision and written opinions, an office name, etc.), etc. [0016] Although a system vendor package-izes these business and supplies it to a user company, as for such business, it is common to be customized so that the environment of a user company, an in-house patent management plan, and the Ruhr may be suited. Drawing 2 (b) The business which attached \* mark is regarded as business with a large user customize part. On the other hand, adjustment with a convention (on these specifications, this is called business logic) that it opted for the input to patent database P-DB in the business environment of the company concerned on these business is required, and this consistency check was shown in a business logic check, a call, and drawing 2 (b). As an example of a business logic check, there is an intellectual property person-in-charge input, or ? and the artifice contribution sum total are [ whether the claim necessity decision less than 100, ? and one year before a request-for-examination term is made, and ] ? etc. Usually, these business logic checks are set as the object of user customize.

[0017] Moreover, since the subordinate of the statute with which object invention is based has these business, the input to patent database P-DB must not be contrary to the logic provided by the statute concerned, and performs a required statute logic check on each business. For example, there is a public disclosure requirement in the application country concerned, or some numbers of \*\* days by ? and the request-for-examination expiration date are ? etc. drawing 2 (b) it is shown -- as -- a statute logic check -- Japanese Patent Law, Japanese Utility Model Law, an United States patent method, and the European Patent Convention (Patent Law of appointed each country) etc. -- although it is a variety and a vendor performs all a support of these, the user is customizing to the system which took in only the country for which its company has applied. In addition, a patent statute needs cautions at every amendment for an application class to increase. For example, it will be 7 by the Heisei 6 amendment of Japanese Patent Law in Heisei 7. Moon 1 Although, as for the patent of application, the duration became 20 years, since defunct law is applied about the application patent before it after a day, a vendor is this 2. A class must be incorporated and the logic which determines the statute applied by the filing date of application of applicable invention must be added.

[0018] Thus, in the conventional system, although each business of a patent managerial system had the element of a business logic check, and the element of a statute logic check, it combined and it was creating these for every user program, so that an operating specification might be suited. drawing 3 (a) \*\*\*\* -- the example of a configuration of the conventional patent managerial system is shown. This is a certain user company (this is called u). It is the patent managerial system customized by turning. In patent management client 1', it is front end check means 11'. Although it is, this is the front end checking program FC (u). It contains. It is FC (u) here. He is User u about the front end checking program FC which a vendor offers. Having customized turning shall be shown. the same -- patent management server 2' \*\*\*\* -- the basis of the user-program selection section -- the open check program GP01 and request-for-examination decision program GP02 (u) the request-for-examination program GP03, the \*\*\*\*\* processing program GP04, and intermediate-processing-intermediate-treatment program GP05 (u) the amendment procedure program GP06, the referee procedure program GP07, and registration-confirmed program GP08 (u) The pension manager GP09 and maintenance necessity decision program GP10 (u) The right expiration processing program GP11 and abandonment processing program GP12 (u) It is constituted. At this example, the open check program GP01 is the request-for-examination decision program GP02 (u), for example, although it is still a vendor offer package. It is the program which customized the vendor offer package.

[0019] The example of a configuration of the request-for-examination decision program GP02 (u) explains this situation to drawing 4 as an example still more concretely. drawing 4 (a) \*\*\*\* -- two users a and b A request-for-examination decision specification is shown. namely, user a \*\*\*\* -- as an inner review classification -- the invention (whenever [ important / a ], b, c) concerned classifying -- ( -- the decision of an amendment important point, a no:claim important point, and unnecessary) is made. And it is the in-house Ruhr that charge \*\*\*\*\* and Importance b at the primary time, and Importance a charges a claim and Importance c at the secondary time. In the case of this user, it is 3 after application at the primary time here. It is 6 after application at the secondary time at the year time. The year time is decided on. On the other hand, he is User b. At a request-for-examination decision specification, it is the invention (an amendment important point, no) concerned as an inner review classification. And (a request-for-examination and reexamination-one year ago time is a hold and claim needlessness instance) is determined.

[0020] The user program GP02 (a) and GP02 (b) reflecting these specifications The example of a configuration becomes like drawing 4 (b). They are routines K1, K2, K3, and K4 here. The core section of GP02 program which a vendor offers is formed, and it is drawing 3 (c). The contents are shown. namely, the routine K1 -- patent management client 1' from -- the reference number of applicable invention and an inner review classification are accepted, and it asks for the filing date of application of applicable invention, and the present date. A routine K2 is the date at the primary time, the date at the secondary time, a request-for-examination expiration date, the number of primary \*\*\*\*\* days, the number of secondary \*\*\*\*\* days, the number of \*\*\*\*\* days, and an expiration date. Expiration date 1 on the previous day of 1 \*\*\*\*\* days etc. are calculated. Here, an application statute is first looked for from filing date of application, it asks for the request-for-examination expiration date provided by the applicable statute, and the dates at the date [ at the primary time defined based on this in the in-house Ruhr ] and secondary time etc. are calculated. In routine K2 design, at the primary time, it devises what kind of time is taken as a secondary point in time so that all users' hope may be filled and many things can be given.

[0021] drawing 4 (b) User a User program GP02 (a) \*\*\*\* -- routines K1 and K2 classification routine [ after

performing the check about a term ] S (a) — the invention concerned — (Importance a, b, and c, claim needlessness, and a term error) — \*\*\*\*\*. This part is User a. It is the routine customized by the in-house specification. Subsequently, the routine K3 (in a term error, a retry is urged and, in a check click, it progresses to a degree) of the core section of GP02 program and a routine K4 (patent data-base-updating means 23' is called, and data base updating is performed) It performs and ends.

[0022] on the other hand — drawing 4 (b) User b User program GP02 (b) \*\*\*\* — routines K1, K2, K3, and K4 a part — user a the same — classification routine S (b) — classification routine S (a) — differing — the invention concerned — (\*\*\*\*\* and one year before — improving — needlessness and term error) — \*\*\*\*\*. This part is User b. It is the routine customized by the in-house specification. It returns to drawing 3 and is drawing 3 (a). It is drawing 3 (b) about the need [ of following on the renewal of a version when there is statute revision in the conventional system / like ] activity. It is shown. A need activity is with renewal activity v-wk of a version, and each customize activity c-wk for users. Since the statute logic check is concerned covering all the business of an offer package as renewal activity v-wk of a version, all the core sections need to be looked again. the case where this is furthermore applied to a user — an applicable user's customize program (the example of drawing 3 (a) — the request-for-examination decision program GP02 (u), the Nakama processing program GP05 (u), the registration-confirmed program GP08 (u), the maintenance necessity decision program GP10 (u), and the abandonment processing program GP12 (u)) It sets and each customize activity c-wk for users is required.

[0023] This invention aims at and has reduction of the activity range accompanying such statute revision, and reduction is expected for the leakage in revised reflection by reduction and a mistake of an activity man day. Request-for-examination decision program GP02 shown by drawing 4 (u) The example in this invention is shown in drawing 5 on the topic of the conventional example. At this invention, it is drawing 5 (a) first. The programming style which improved the conventional user program and separated the business logic part inherent and the statute logic part is taken so that it may be shown. Specifically at the example of drawing 4, they are routines K1 and K3. Although it is the business logic section and a routine K4 is the statute logic section since statute logic is not included, they are a routine K2 and classification routine S (a), and S (b). Since it sets and the statute logic part is intermingled, it dissociates, as shown in drawing.

[0024] That is, it is routine K2L about the routine which calculates the expiration date which is a statute logic part from a routine K2. It dissociates by carrying out and is the remainder Business logic routine K2B It carries out. It is expressed  $K2=K2B+K2L$  in graph. Similarly, it is classification routine  $S=$  classification business logic routine  $SB+$  Statute logic routine  $SL$  It dissociates. namely, a graph —like —  $S(a)=SB(a)+SL$  and  $S(b)=SB(b)+SL$  it is . It is the part for which it does not depend on a user specification here although  $SL$  is a statute logic check part common to a user about classification and it depends for it on a statute.

[0025] Thus, reduction of the renewal range of a version at the time of statute revision is achieved by separating a business logic check part and a statute logic check part, and creating a program. Drawing 6 explains this. Namely, drawing 6 (a) The patent managerial system of this invention separates and has the business logic check means 21 (program BC (u)) and the statute logic check means 22 (program LC (u)) like. When it is made such a configuration, the renewal range of a version at the time of statute revision is drawing 6 (b). It comes to be shown. this — drawing 3 (b) of the conventional example if it compares — general —  $LC$  core section size  $< (GP01+GP02+ \dots +GP12)$

Since core section size is materialized, reduction of the range of renewal activity v-wk of a version in a vendor is achieved, and reduction in a workload is achieved. moreover — the customize section — general —  $LC(u)$  Size  $< (GP01(u)+GP02(u) + \dots +GP12(u))$  Since it is size  $(LC(a) + \dots +LC(e)) (+ \dots [ It is + (GP01(e)+GP02(e) + \dots +GP12(e))$  size. ]) Size  $< — (GP01(a)+GP02(a) + \dots +GP12(a))$  Therefore, reduction of each customize activity c-wk load for users in accordance with statute revision is also achieved.

[0026] It returns to drawing 5 again and is drawing 5 (b) about the structure of each customize for users of this invention. It explains. user a turning — user program GP02 (a) the program operated substantially — drawing 4 and drawing 5 (a) from — K1, K2B, and K2 — L, SB (a), and SL, K3 and K4 it is — things are understood. Among these, K1, K2B, SB (a), and K3 It is contained in the business logic check means 21, and is K2L, and SL and K4. It is as having already stated to be contained in the statute logic check means 22. K1 in the business logic check means 21, K2B, SB (a), and K3 When its attention is paid, it is K1, K2B, and K3. It is the core section and is SB (a). User a It turns and is the customize section. In this invention, these routines are implemented as an object set with an object-oriented technique. And it realizes customizing a core object using the succession relation between objects.

[0027] Drawing 5 (b) The structure is explained. At this invention, it is drawing 5 (b). The business logic check means 21 is divided into business logic check customize section BC-C and business logic check common section BC-K like. Moreover, it separates into statute logic check customize section LC-C and statute logic check common section LC-K also about the statute logic check means 22. At this time, the object which builds in the effective method offered by the vendor is made into a abstract class, this is made into the common section (for example, business logic check common section BC-K, statute logic check common section LC-K), and the activation class as an interface which performs this is offered as the customize section (for example, business logic check customize section BC-C, statute logic check customize section LC-C). Thus, when a method is not built in but a start up message is received itself, the object of the offered activation class follows the abstract class which is a parent class, and performs the method of the object of relevance. And when customize is required, the activation class object of this customize section is transposed to the customize object which described the contents of customize as a method.

[0028] He is User a about the above relation. It turns and is a user program GP02 (a). When it explains by the case,

it is drawing 5 (b). In business logic check common section BC-K, it is K2L, and SL and K4 like to K1, K2B, K3 and SB, and statute logic check common section LC-K. The abstract class object is prepared, and — these — corresponding — activation — a class — an object — K — one — ' — K — 2B — ' — K — three — ' — SB — ' — K — two — L — ' — SL — ' — K — four — ' — customize — the section — preparing — having — \*\*\*\*. On the other hand, user a Classification routine SB+SL of the common section SB of a abstract class is not changed to change into SB(a)+SL with a user specification, but it is SB' of an activation class. SB (a) It replaces.

[0029] He is User a by this. User b Having business logic check common section BC-K common to a system, and statute logic check common section LC-K, a vendor is common business logic check common section BC-K of one \*\* to all users, What is necessary is just to support statute logic check common section LC-K. And since business logic check common section BC-K has the structure where it is not influenced of statute revision, if in charge of statute revision, a vendor should just perform renewal activity v-wk of a version to statute logic check common section LC-K of one \*\*. Moreover, with this structure, the contents of statute revision are automatically reflected in the user customize section with the succession relation of an object.

[0030] When drawing 7 repeats the above explanation, it is drawing 7 R> 7 (a). In the system of an example, the business logic check means 21 of the patent management server 2 is the structure divided into business logic check customize section BC-C and business logic check common section BC-K, and has divided the statute logic check means 22 into statute logic check customize section LC-C and statute logic check common section LC-K. The range of the renewal activity of a statute revised version in this condition is drawing 7 (b). Since it is limited to front end check common section FC-K, business logic check common section BC-K, and statute logic check common section LC-K so that it may be shown, reduction of the range of renewal activity v-wk of a version is achieved conventionally, and reduction in a workload is achieved. Furthermore, FC-C (u), BC-C (u), and LC-C (u) FC-K, BC-K, and LC-K Since it is the activation class which inherits the contents of renewal activity v-wk of a version, modification accompanying statute revision has needlessness, therefore an unnecessary user customize man day.

[0031] Furthermore, drawing 8 explains the 3rd example. The patent managerial system of drawing 8 (a) is the structure which separated statute database L-DB further in the statute logic check means 22. Statute database L-DB is constituted as a set of an item which was suitable in order to make a computer carry out the logic check of the predetermined description of a statute, and shows the example of structure to drawing 10.

[0032] At the example of drawing 10, it is drawing 10 (a). Statute database L-DB is the set of the database file (Japanese Patent Law .db and United States patent method .db, ...) corresponding to the statute according to individual so that it may be shown. For example, the structure of statute database L-DB corresponding to Japanese Patent Law is drawing 10 (b). All past revision is included so that it may be shown. Drawing is an example at the Heisei 6 amendment time. As shown in drawing, an item required for a statute logic check is arranged by a large item, a mean eye, and the subparagraph, each item accepts the need, and it is double \*\*\*\*\*. For example, since it is eternal even if it is 7 year" from "filing date of application from the Heisei 5 amendment or before and will be in amendment about a request-for-examination term in Heisei 6, this column is "ap. — It is 1 filled in as 07y00m00d". Only a cel is required. However, if it is "filing-date-of-application < 2001 year 10 month 1 day when amendment will be reflected in Heisei 11 since it will become 3 year" from "filing date of application by amendment Heisei 11 and is 7 year" and "filing-date-of-application > 2001 year 10 month 1 day from filing date of application, it is 3 from filing date of application. 2 of year" \*\* must be chosen and applied. When this is reflected on statute database L-DB, they are "application Ruhr:ap, 011001, and term:ap. — 07y00m00d" and "application Ruhr:011001, ap, term:ap — 2 of 03y00m00d" Item 2 It will be rewritten by the train.

[0033] When it returns to drawing 8, it is drawing 8 (a). In the patent managerial system constituted like, as shown in drawing 8 (b), only statute database L-DB instead of an object of the renewal activity of a statute logic check common section LC-K version can be made into the range of renewal activity v-wk of a version in statute revision. Although this is revision of the range which does not change the framework of a statute, for example, the request-for-examination system itself is lost, or it is assumed that the renewal activity of a version is needed also for statute logic check common section LC-K when it is the large revision in which a new procedure is inserted, to the track record of the framework change by the present, it can be coped with by modification of statute database L-DB.

[0034] Next, modification of statute database L-DB accompanying statute revision is considered. In this invention, the statute data-base-updating server 3 which updates statute database L-DB as shown in drawing 1 is proposed. Next, drawing 12 is used together and the example of the statute data-base-updating server 3 of operation is explained. It is drawing 12 (a) about processing at the time of renewal of revised statute registration. A flow of operation explains. The statute data-base-updating server 3 has the revised statute registration means 32, and the revised statute registration means 32 is revised statute database information L-DB' by step S1201. It reads and stores in the statute data-base-updating server 3. Revised statute database information L-DB' Drawing 10 (b) It is created by the vendor engineer as information which rewrites only a part for the data division changed so that it may be suitable for structure, and is supplied to the revised statute registration means 32. Subsequently, the e-mail means which the statute data-base-updating server 3 does not illustrate extracts the user who is using the statute of revision from the support user list in step S1202 this time with operator directions. Whenever a patent managerial system is supplied to a user from a vendor, renewal of record of this support user list is carried out at the statute data-base-updating server 3 with that customize information, a user person-in-charge mail address, a user person-in-charge mail address, etc. Subsequently, the e-mail means which the statute data-base-updating server 3 does not illustrate creates updating guidance mail to these users at step S1204, and makes it a user person-in-charge

mail address with automatic feed while it generates a non-receiving-user list and prints the user who determined at step S1202 as a user for transmission at step S1203. The contents of updating guidance tell for example, a revised outline, an application beginning date, the download approach, the install approach, etc.

[0035] When the user person in charge who received the above-mentioned mail sends a download demand to the homepage of the statute data-base-updating server 3, it is drawing 12 (c). Processing begins at the time of download demand reception. The revised statute database information transmitting means 31 of the statute data-base-updating server 3 checks a download demand user by step S1209, and it checks that he is a right user and that the privacy of contract of download costs is processing settled by authentication processing. If the result is judged at step S1210 and it is judged with transmitting authorization, the revised statute database information transmitting means 31 will be the revised statute database information L-DB' concerned at step S1211. It transmits. And if transmission is successful, a transmitting settled one will be recorded on a non-receiving-user list, an erase lump will be performed, and a non-receiving-user list is updated.

[0036] furthermore, the common control section which the statute data-base-updating server 3 does not illustrate — the operator appointed term period — every day — scheduled time — drawing 12 (b) it is shown — Japanese — degree processing is started. That is, the alarm point judging means 33 of the statute data-base-updating server 3 expresses the non-receiving-user list of [ at the current times ] as step S1205, calculates the number of \*\* days by the revised statute application day concerned at step S1206, and if the number of \*\* days is 15 days about step S1207 when the number of \*\* days is 30 days, it will start step S1208.

[0037] At step S1207, the notice means 34 of an alarm of the statute data-base-updating server 3 carries out alarm mail to the person in charge and person in charge of the non-receiving user concerned with automatic feed. It is area-of-influence [ when not applying the statute revision concerned as the contents ], responsibility matter, and revised statute database information L-DB'. Install, test necessary period anticipation, etc. are shown and the reply of a reply is requested. In addition, although not illustrated to drawing 12, when it is checked that it may not receive by the above-mentioned reply, the operator of the statute data-base-updating server 3 deletes the user concerned from a non-receiving-user list by the manual. At step S1208, the notice means 34 of an alarm carries out contact request mail to the operating person in charge of the vendor which supports a non-receiving user in charge, and a operating person in charge with automatic feed. The contents of e-mail are requested to contact the user concerned and to take an immediate action.

[0038] To the statute data-base-updating server 3 explained above, at the example of the patent managerial system of this invention shown in drawing 9, the patent management server 2 has the renewal means 24 of statute database reception, and is revised statute database information L-DB' from said statute data-base-updating server 3. It receives and statute database L-DB is overwritten. At such an example, it is revised statute database information L-DB'. Activation of renewal activity v-wk of a version which updates statute database L-DB is attained to a user site. Therefore, a user does not need to pay updating package install activity cost to a vendor like before.

[0039] It is revised statute database information L-DB' by the XML format although it is arbitrary about the data format of revised statute database information L-DB', as shown in drawing 1111 as an example of this invention. The example of an expression is explained. This format is drawing 10 (b). It corresponds to DS, each subject name of a large item, a mean eye, and a subparagraph is used as a tag, and a data value is initiation/. It is indicated between termination tags.

[0040] Drawing 11 (a) Heisei 6 It is the example of an extract of revised statute database information L-DB' at the time of year amendment, and is this revised statute database information L-DB'. The condition of having been stored is drawing 10 (b). It corresponds to an expiration Ruhr part. Moreover, drawing 10 of the Heisei 11 amendment explained previously (b) Revised statute database information L-DB' corresponding to \* mark An extract is drawing 11 (b). It is shown. ap, pb, rg, and or in data It is the variable name which shows filing date of application, a public notice day, a registration day, and criteria filing date of application, respectively, for example, is drawing 1111 (a). The three expiration Ruhr A, B, and C is expressing the respectively following Ruhr.

Expiration Ruhr A: Start this Ruhr by action which inputs a public notice day into patent database P-DB, and if it is on filing-date-of-application < January 1, 1994, set up 15 years with right expiration from a public notice day.

Expiration Ruhr B: Start this Ruhr by action which inputs a public notice day into patent database P-DB, and if it is on January 1, 1994 < filing-date-of-application < July 1, 1995, set up the period of the shorter one with right expiration among 15 years from a public notice day for 20 years from filing date of application.

Expiration Ruhr C: Start this Ruhr by action which inputs a registration day into patent database P-DB, and if it is July 1, 1995 < filing date of application, set up 20 years with right expiration from criteria filing date of application.

[0041] It can rewrite by adopting such a format and only a need item can be transmitted in intelligible form.

Moreover, also except the patent managerial system of this invention, such opening formal data can be applied as data, and, generally have the circulation possibility as charged contents.

(Additional remark 1) The statute dependence housing management server which manages the procedure which has the statute dependence housing database which stores the tendency hysteresis of a statute dependence housing, and is needed based on a statute, In the statute dependence housing managerial system which consists of one or more statute dependence housing management clients which perform the data registration to said statute dependence housing database, and an updating input It is a statute dependence housing manager for making the computer of a statute dependence housing management server perform statute dependence housing management information processing. As opposed to the data input for the data registration to said statute dependence housing database from said statute dependence housing management client, and updating It checks according to the



business logic which is the Ruhr in management business appointed beforehand. The business logic checking program which receives only the data which will stimulate reinput if there is an error, and do not have the error as business logic and as which a computer is operated as a business logic check means, To said data input, if there is an error in the light of the statute logic which is the Ruhr in a statute appointed beforehand, reinput will be urged. The statute logic checking program which receives only data without a statute logic error and as which a computer is operated as a statute logic check means, The statute dependence housing manager characterized by \*\*\*\*(ing) and said business logic checking program and said statute logic checking program having dissociated.

[0042] (Additional remark 2) Said business logic checking program Furthermore, the business logic check common section which is the abstract class object set which describes business logic check processing in common, The business logic check customize section customized so that it might consist of an activation class for performing said business logic check common section and the system environment and the operating conditions for every user might be suited, It is the program boiled and separated. Said statute logic checking program Furthermore, the statute logic check common section which is the abstract class object set which describes statute logic check processing in common, The statute logic check customize section customized so that it might consist of an activation class for performing said statute logic check common section and the system environment and the operating conditions for every user might be suited, The statute dependence housing manager of the additional remark 1 publication which is alike and is characterized by being the separated program.

[0043] (Additional remark 3) Said statute logic check common section is the statute dependence housing manager of the additional remark 2 publication characterized by be the structure of have outside the statute database constituted as a set of an item which was suitable further in order to make a computer carry out the logic check of the predetermined description of a statute, and perform a statute logic check with reference to the statute database concerned.

[0044] (Additional remark 4) The statute dependence housing management server which manages the procedure which has the statute dependence housing database which stores the tendency hysteresis of a statute dependence housing, and is needed based on a statute, It is the statute dependence housing managerial system which consists of one or more statute dependence housing management clients which perform the data registration to said statute dependence housing database, and an updating input. A front end check means to receive only the data which perform the format check of the data input for the data registration to said statute dependence housing database, and updating, will stimulate reinput if there is an error, and do not have a format error, The statute dependence housing managerial system characterized by having the statute dependence housing management server which has the business logic check means and the statute logic check means of functioning by the statute dependence housing manager additional remarks 1 or 2 or given in three.

[0045] (Additional remark 5) Statute data-base-updating server which has a revised statute registration means register as revised statute database information constituted as a set of an item which was suitable in order to make a computer carry out the logic check of the predetermined description of the revised statute, and a revised statute database information transmitting means transmit said revised statute database information through the Internet to a statute dependence housing managerial system.

[0046] (Additional remark 6) Are the statute data-base-updating server of additional remark 5 publication, and the customize information, said revised statute database information, the current time, and the transmitted hysteresis of said statute dependence housing managerial system are contrasted further. An alarm point judging means to extract the statute dependence housing managerial system which has not been received although said revised statute database information needs to be received, The statute data-base-updating server characterized by having a notice means of an alarm to perform the notice of an alarm to the statute dependence housing managerial system which said alarm point judging means extracted.

[0047] (Additional remark 7) Statute dependence housing managerial system characterized by having a renewal means of statute database reception to receive revised statute database information through the Internet from a statute data-base-updating server additional remark 5 or given in six, and to update a statute database.

(Additional remark 8) The statute dependence housing management server which manages the procedure which has the statute dependence housing database which stores the tendency hysteresis of a statute dependence housing, and is needed based on a statute, It is the statute dependence housing management method of the statute dependence housing managerial system which consists of one or more statute dependence housing management clients which perform the data registration to said statute dependence housing database, and an updating input. The front end check phase of receiving only the data which perform the format check of the data input for the data registration to said statute dependence housing database, and updating, will stimulate reinput if there is an error, and do not have a format error, As opposed to the data input for the data registration to said statute dependence housing database from said statute dependence housing management client, and updating The business logic check phase of receiving only the data which check according to the business logic which is the Ruhr in management business appointed beforehand, will stimulate reinput if there is an error, and do not have the error as business logic. The statute dependence housing management method characterized by having the statute logic check phase of receiving only the data which will stimulate reinput if there is an error in the light of the statute logic which is the Ruhr in a statute appointed beforehand to said data input, and do not have a statute logic error.

[0048] (Additional remark 9) Record medium which recorded the statute dependence housing manager additional remarks 1 or 2 or given in three and in which computer reading is possible.

(Additional remark 10) It is the statute data-base-updating server program which makes a statute data-base-

updating server perform information processing which supplies revised statute database information to a statute dependence housing managerial system. The revised statute registration phase registered as revised statute database information constituted as a set of an item which was suitable in order to make a computer carry out the logic check of the predetermined description of the revised statute. The statute data-base-updating server program characterized by making a statute data-base-updating server perform the revised statute database information transmitting phase of transmitting said revised statute database information through the Internet to a statute dependence housing managerial system.

[0049] (Additional remark 11) Are the statute data-base-updating server program of additional remark 10 publication, and the customize information, said revised statute database information, the current time, and the transmitted hysteresis of said statute dependence housing managerial system are contrasted further. The alarm point judging phase of extracting the statute dependence housing managerial system which has not been received although said revised statute database information needs to be received. The statute data-base-updating server program characterized by making a statute data-base-updating server perform the notice phase of an alarm of performing the notice of an alarm to the statute dependence housing managerial system which said alarm point judging means extracted.

[0050] (Additional remark 12) Statute dependence housing manager characterized by making a computer perform the renewal phase of statute database reception which is a statute dependence housing manager for making a statute dependence housing management system computer perform statute dependence housing management, receives revised statute database information through the Internet from a statute data-base-updating server additional remark 5 or given in six, and updates a statute database.

[0051]

[Effect of the Invention] According to this invention, it is effective in offer of the patent managerial system which decreases more the version up range of the package accompanying statute revision, and decreases the leakage in revised reflection by reduction and a mistake of an activity man day being attained so that clearly from the above explanation. Moreover, it is effective in offer of the patent managerial system which makes the activity man day of the user customize accompanying version up of a package reduce being attained. Moreover, although the example of this invention explained the detailed contents in a patent managerial system, it is being able to say the statute dependence housing managerial system at large which manages not only a patent managerial system but the statute dependence housing which generally performs the procedure which is needed based on a statute.

---

[Translation done.]

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2003-76816

(P 2 0 0 3 - 7 6 8 1 6 A)

(43) 公開日 平成15年3月14日 (2003. 3. 14)

|                            |      |            |     |            |
|----------------------------|------|------------|-----|------------|
| (51) Int. Cl. <sup>7</sup> | 識別記号 | F I        |     | テマコード (参考) |
| G06F 17/60                 | 154  | G06F 17/60 | 154 |            |
| 9/44                       | 560  | 9/44       | 560 | F          |

審査請求 未請求 請求項の数 9 O L (全15頁)

(21) 出願番号 特願2001-264993 (P 2001-264993)

(22) 出願日 平成13年8月31日 (2001. 8. 31)

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 発明者 家富 慎二

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74) 代理人 100072590

弁理士 井桁 貞一

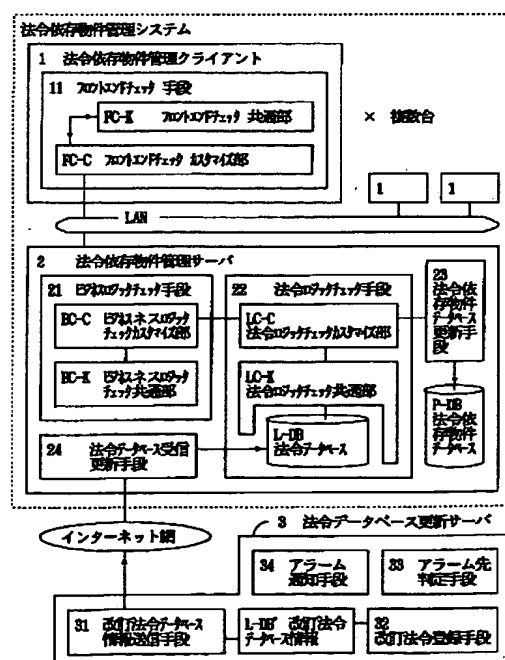
最終頁に続く

(54) 【発明の名称】 法令依存物件管理プログラム、法令依存物件管理システム、および法令データベース更新サーバ

## (57) 【要約】

【課題】 従来の法令依存物件管理システム（例えば特許管理システム）では法令改訂のたびにベンダは正確迅速なシステムパッケージのバージョンアップのために多大な工数が必要であり、ユーザもこのバージョンアップに伴うカスタマイズに多大な工数を要するという課題があった。

【解決手段】 法令依存物件管理プログラムをビジネスロジックチェックプログラムと法令ロジックチェックプログラムとに分離、更に、法令ロジックチェックプログラムを法令ロジックチェック共通部と法令ロジックチェックカスタマイズ部とに分離、更に法令ロジックチェックプログラムから法令データベース部を分離すると、法令改訂でのバージョンアップ範囲を法令ロジックチェックプログラムに、更に法令ロジックチェック共通部に、更に法令データベース部に限定でき改訂ミスを減少し工数削減ができる。

本発明の法令依存物件管理システムおよび  
法令データベース更新サーバの構成例

## 【特許請求の範囲】

【請求項 1】 法令依存物件の帰趨履歴を格納する法令依存物件データベースを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバと、前記法令依存物件データベースへのデータ登録、更新入力を行う 1 つ以上の法令依存物件管理クライアントとから構成される法令依存物件管理システムにおいて、法令依存物件管理サーバのコンピュータに法令依存物件管理情報処理を実行させるための法令依存物件管理プログラムであって、

前記法令依存物件管理クライアントからの前記法令依存物件データベースへのデータ登録、更新のためのデータ入力に対して、管理業務における予め定められたルールであるビジネスロジックに従ってチェックを行い、エラーがあれば再入力を促し、ビジネスロジックとしてのエラーのないデータのみを受け入れるビジネスロジックチェック手段としてコンピュータを機能させるビジネスロジックチェックプログラムと、

前記データ入力に対して、法令における予め定められたルールである法令ロジックに照らしてエラーがあれば再入力を促し、法令ロジックエラーのないデータのみを受け入れる法令ロジックチェック手段としてコンピュータを機能させる法令ロジックチェックプログラムと、を有し、前記ビジネスロジックチェックプログラムと前記法令ロジックチェックプログラムとが分離していることを特徴とする法令依存物件管理プログラム。

【請求項 2】 前記ビジネスロジックチェックプログラムは、さらに、

ビジネスロジックチェック処理を共通的に記述する抽象クラスオブジェクト集合であるビジネスロジックチェック共通部と、

前記ビジネスロジックチェック共通部を実行するための実行クラスからなりユーザ毎のシステム環境や業務条件に適合するようにカスタマイズされたビジネスロジックチェックカスタマイズ部と、に分離されたプログラムであり、

前記法令ロジックチェックプログラムは、さらに、法令ロジックチェック処理を共通的に記述する抽象クラスオブジェクト集合である法令ロジックチェック共通部と、

前記法令ロジックチェック共通部を実行するための実行クラスからなりユーザ毎のシステム環境や業務条件に適合するようにカスタマイズされた法令ロジックチェックカスタマイズ部と、に分離されたプログラムであることを特徴とする請求項 1 記載の法令依存物件管理プログラム。

【請求項 3】 前記法令ロジックチェック共通部は、さらに、

法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した法令データ

ベースを外部に有し、当該法令データベースを参照して法令ロジックチェックを行う構造であることを特徴とする請求項 2 記載の法令依存物件管理プログラム。

【請求項 4】 法令依存物件の帰趨履歴を格納する法令依存物件データベースを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバと、前記法令依存物件データベースへのデータ登録、更新入力を行う 1 つ以上の法令依存物件管理クライアントとから構成される法令依存物件管理システムであって、

10 前記法令依存物件データベースへのデータ登録、更新のためのデータ入力のフォーマットチェックを行い、エラーがあれば再入力を促し、フォーマットエラーのないデータのみを受け入れるフロントエンドチェック手段と、請求項 1 または 2 または 3 記載の法令依存物件管理プログラムによって機能するビジネスロジックチェック手段および法令ロジックチェック手段を有する法令依存物件管理サーバと、を有することを特徴とする法令依存物件管理システム。

【請求項 5】 改訂された法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した改訂法令データベース情報として登録する改訂法令登録手段と、法令依存物件管理システムに対してインターネットを通じて前記改訂法令データベース情報を送信する改訂法令データベース情報送信手段とを有する法令データベース更新サーバ。

【請求項 6】 請求項 5 記載の法令データベース更新サーバであって、さらに、

前記法令依存物件管理システムのカスタマイズ情報と前記改訂法令データベース情報と現在日時および送信履歴とを対比して、前記改訂法令データベース情報の受信が必要であるのに未受信である法令依存物件管理システムを抽出するアラーム先判定手段と、

前記アラーム先判定手段が抽出した法令依存物件管理システムに対してアラーム通知を行うアラーム通知手段と、を有することを特徴とする法令データベース更新サーバ。

【請求項 7】 請求項 5 または 6 記載の法令データベース更新サーバからインターネットを通じて改訂法令データベース情報を受信し法令データベースを更新する法令データベース受信更新手段を有することを特徴とする法令依存物件管理システム。

【請求項 8】 法令依存物件の帰趨履歴を格納する法令依存物件データベースを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバと、前記法令依存物件データベースへのデータ登録、更新入力を行う 1 つ以上の法令依存物件管理クライアントとから構成される法令依存物件管理システムの法令依存物件管理方法であって、

前記法令依存物件データベースへのデータ登録、更新のためのデータ入力のフォーマットチェックを行い、エラ

10

20

30

40

50

ーがあれば再入力を促し、フォーマットエラーのないデータのみを受け入れるフロントエンドチェック段階と、前記法令依存物件管理クライアントからの前記法令依存物件データベースへのデータ登録、更新のためのデータ入力に対して、管理業務における予め定められたルールであるビジネスロジックに従ってチェックを行い、エラーがあれば再入力を促し、ビジネスロジックとしてのエラーのないデータのみを受け入れるビジネスロジックチェック段階と、前記データ入力に対し法令における予め定められたルールである法令ロジックに照らしてエラーがあれば再入力

を促し、法令ロジックエラーのないデータのみを受け入れる法令ロジックチェック段階と、を有することを特徴とする法令依存物件管理方法。

【請求項 9】 請求項 1 または 2 または 3 記載の法令依存物件管理プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、技術開発企業等において自社出願特許の帰趨を管理する特許管理システムのように、法令依存物件（例えば出願済特許）を管理する法令依存物件管理システムの構築技術に係り、特に、法令の改訂に伴うシステムレベルアップを効率よく行う技術に関する。

【0002】

【従来の技術】世界的な技術開発競争時代にはいり、特に技術開発企業においては特許戦略が経営戦略の重要な一部となってきた。世界をマーケットとする技術開発企業においては、日本のみならず米国、欧州、アジアの各国に対し戦略的に特許出願を行い、適切な権利取得を図っている。このように多数の出願件数をもつ企業では、特許戦略推進のために、IT 技術を用いて特許管理業務のシステム化を図ってきた。図 2 (a) にはこのような特許管理業務のシステム化の全貌を示す。

【0003】図に示す如く、代表的なシステムとして、

1. 公知例検索システム：CD-ROM 公報の蓄積、検索機能を持ち、発明者や社内審査部門が自社特許アイデアを検討する際に使用する。
2. 発明者出願支援システム：社内出願依頼、パソコン出願、事務所連携などの機能を持ち、出願実務をサポートする。
3. 特許管理システム：発明者からの出願依頼の社内受付、進捗管理、年金管理、経費管理等を行い、出願特許の帰趨履歴を管理する。
4. 包袋管理システム：出願書類、手続き書類の電子保管、検索機能を持ち、問題特許や特許係争時のツールとなる。

【0004】本発明は、上記範囲のうち、3. 特許管理システムに関するものである。特許管理システムは特許誕

生以前から権利満了に至る特許のライフサイクル中の帰趨履歴を特許データベースで管理する。システムには特許ライフサイクルに則して多くの特許管理業務が含まれるが、図 2 (b) にその代表的な業務一覧を示す。本発明はそのうち、特許管理システムの特許データベース P-DB を更新する特許管理業務に関するもので、図 2 (b) で◎を付した部分であるが、これについては後に説明する。

【0005】多数の技術開発企業などでこれらの特許管理システムを導入しているが、その多くはシステムベンダが提供する特許管理システムパッケージを購入し、自社環境や社内特許管理方針やルールに適合するようにカスタマイズを行ってこのシステムを運用している。しかるに、特許権をはじめとする工業所有権を定める法令（以下特許法令と呼ぶ）は多くは各国ごとに異なり、特許権、商標権など種類が多いために、大規模な特許管理システムユーザでは 30 国以上、100 種以上の特許法令を取り入れている。システムベンダはもちろんこれらすべてをサポートする。

【0006】さらに、プロパテント時代を反映して、これら法令は小改訂や大改訂をとりまぜて頻繁に改訂されている。例えば、日本特許法では近年でも平成 5 年改正、平成 6 年改正、平成 10 年改正、平成 11 年改正、と続いている。しかも、現行日本特許期間が出願から 20 年であるように、一般に特許管理は長期管理を要するので、何世代も前の特許法を適用する事件特許の管理が必要である。

【0007】以上述べた特許管理システムのように、多数の法令に依存する物件（例えば出願済特許）を管理する法令依存物件管理システムでは、上記のように多種の関連する法令を抱えていると年 1～2 回の改訂に遭遇し、システムベンダではその改訂のつどパッケージのバージョンアップ作業への没頭が必要であり、さらに、バージョンアップシステムをユーザに適用するにあたって、当該ユーザのカスタマイズ箇所をユーザ毎に把握して、当該バージョンアップに整合するように新たなカスタマイズ作業が必要であった。

【0008】

【発明が解決しようとする課題】このように従来の特許管理システムに代表されるような法令依存物件管理システムでは、法令改訂のたびにベンダーは正確迅速なシステムパッケージのバージョンアップのために多大な工数が必要であり、ユーザもこのバージョンアップに伴うカスタマイズに多大な工数を要するという課題があった。本発明はこのような課題を解決し、法令改訂に伴うパッケージのバージョンアップ範囲をより減少させ、作業工数の削減とミスによる改訂反映漏れを減少させる法令依存物件管理システムを提供することを目的とする。また、パッケージのバージョンアップに伴うユーザカスタマイズの作業工数を削減させる法令依存物件管理システムを提供することを目的とする。

## 【0009】

【課題を解決するための手段】上記課題は、図1に示す如く、法令依存物件の帰趨履歴を格納する法令依存物件データベースP-DBを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバ2と、前記法令依存物件データベースP-DBへのデータ登録、更新入力を行う1つ以上の法令依存物件管理クライアント1とから構成される法令依存物件管理システムにおいて、法令依存物件管理サーバ2のコンピュータに法令依存物件管理情報処理を実行させるための法令依存物件管理プログラムであって、前記法令依存物件管理クライアント1からの前記法令依存物件データベースP-DBへのデータ登録、更新のためのデータ入力に対して、管理業務における予め定められたルールであるビジネスロジックに従ってチェックを行い、エラーがあれば再入力を促し、ビジネスロジックとしてのエラーのないデータのみを受け入れるビジネスロジックチェック手段21としてコンピュータを機能させるビジネスロジックチェックプログラムと、前記データ入力に対し法令における予め定められたルールである法令ロジックに照らしてエラーがあれば再入力を促し、法令ロジックエラーのないデータのみを受け入れる法令ロジックチェック手段22としてコンピュータを機能させる法令ロジックチェックプログラムと、を有し、前記ビジネスロジックチェックプログラムと前記法令ロジックチェックプログラムとが分離している法令依存物件管理プログラム、によって解決される。

## 【0010】また前記ビジネスロジックチェック機能

(ビジネスロジックチェック手段21の機能)はビジネスロジックチェック処理を共通的に記述する抽象クラスオブジェクト集合であるビジネスロジックチェック共通部BC-Kと、前記ビジネスロジックチェック共通部BC-Kを実行するための実行クラスからなりユーザ毎のシステム環境や業務条件に適合するようにカスタマイズされたビジネスロジックチェックカスタマイズ部BC-Cとから構成されたプログラムによって実現され、前記法令ロジックチェック機能(法令ロジックチェック手段22の機能)は法令ロジックチェック処理を共通的に記述する抽象クラスオブジェクト集合である法令ロジックチェック共通部LC-Kと、前記法令ロジックチェック共通部LC-Kを実行するための実行クラスからなりユーザ毎のシステム環境や業務条件に適合するようにカスタマイズされた法令ロジックチェックカスタマイズ部LC-Cとから構成されたプログラムによって実現する法令依存物件管理プログラム、によって上記課題は解決される。

【0011】また、法令ロジックチェック機能(法令ロジックチェック手段22の機能)は法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した法令データベースL-DBを、法令ロジックチェック共通部LC-Kと分離して保持する法令依存物件管理プログラム、によって上記課題は解決される。

【0012】また、改訂された法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した改訂法令データベース情報として登録する改訂法令登録手段32と、法令依存物件管理システムに対してインターネットを通じて前記改訂法令データベース情報を送信する改訂法令データベース情報送信手段31とを有する法令データベース更新サーバ3から、インターネットを通じて改訂法令データベース情報を受信し法令データベースL-DBを更新する法令データベース受信更新手段24を有する法令依存物件管理システム、によって上記課題は解決される。

【0013】以上の作用については実施例に則して以下に詳述する。

## 【0014】

【発明の実施の形態】本発明の法令依存物件管理プログラム、法令依存物件管理システム、および法令データベース更新サーバの実施例を図1～図12により説明する。なお、本発明におけるコンピュータ処理は、コンピュータプログラムにより当該コンピュータの主記憶装置上で実行されるが、このコンピュータプログラムの提供形態は、当該コンピュータに接続された補助記憶装置をはじめ、CD-ROM等の可搬型記憶装置やネットワーク接続された他のコンピュータの主記憶装置及び補助記憶装置等の各記録媒体に格納されて提供されるもので、このコンピュータプログラムの実行に際しては、当該コンピュータの主記憶装置上にローディングされ実行されるものである。

【0015】先に図2(b)で説明したように、特許データベースP-DBを更新する特許管理業務は多様である。以下にその業務概要を抜粋して示すと、公開確認(出願特許の公開番号、公開日入力)、審査請求判断(社内審査による出願特許の審査請求要否、請求時期、補正有無の判断結果入力)、審査請求(請求日、ほか事務データ入力)、庁通知処理(拒絶理由通知等、各種通知受理入力)、中間処理(社内担当部門による庁への対応アクション判断結果の入力)、補正手続(補正書、意見書の発送日、事務所名等の入力)などである。

【0016】システムベンダはこれら業務をパッケージ化してユーザ企業に納入するが、これらの業務はユーザ企業の環境や社内特許管理方針やルールに適合するようにカスタマイズされるのが普通である。図2(b)で※印を付した業務はユーザカスタマイズ部分が大きい業務と見られるものである。一方、これら業務で特許データベースP-DBへの入力は当該企業のビジネス環境で決められた約束(本明細書ではこれをビジネスロジックと呼ぶ)との整合性が必要であり、この整合性チェックをビジネスロジックチェックと呼び、図2(b)に示した。ビジネスロジックチェックの例としては、知財担当者入力があるか?、発明者寄与率合計が100以内か?、審査請求期限一年前の請求要否判断がなされているか?、などで

ある。通常これらのビジネスロジックチェックはユーザカスタマイズの対象となる。

【0017】また、これら業務は対象発明の依拠する法令の配下にあるので、特許データベースP-DBへの入力は当該法令の定めるロジックに反するものであってはならず、各業務では必要な法令ロジックチェックを行う。例えば、当該出願国には公開制度があるか？、審査請求期限日までの残日数はいくらか？、などである。図2(b)に示す如く、法令ロジックチェックは日本特許法、日本実用新案法、米国特許法、欧州特許条約(指定各国特許法)など多種であり、ベンダはこれらすべてのサポートを行うが、ユーザは自社が出願している国だけを取り入れたシステムにカスタマイズしている。なお、特許法令は改正の都度、適用種類が増えることに注意が必要である。例えば日本特許法の平成6年改正で平成7年7月1日以降出願の特許は存続期間が20年となったが、それ以前の出願特許については旧法が適用されるので、ベンダはこの2種類とも取り込み、該当発明の出願日によって適用する法令を決定するロジックを付加しなければならない。

【0018】このように特許管理システムの各業務はビジネスロジックチェックの要素と法令ロジックチェックの要素を持つが、従来のシステムでは各業務プログラムごとにこれらを業務仕様に適合するように組み合わせて作成していた。図3(a)には従来の特許管理システムの構成例を示す。これはあるユーザ企業(これをuと呼ぶ)向けにカスタマイズされた特許管理システムである。特許管理クライアント1'にはフロントエンドチェック手段11'があるが、これはフロントエンドチェックプログラムFC(u)を含んでいる。ここにFC(u)はベンダの提供するフロントエンドチェックプログラムFCをユーザu向けにカスタマイズしたことを示すものとする。同様に特許管理サーバ2'には業務プログラム選択部のもとに、公開確認プログラムGP01、審査請求判断プログラムGP02(u)、審査請求プログラムGP03、庁通知処理プログラムGP04、中間処理プログラムGP05(u)、補正手続プログラムGP06、審判手続プログラムGP07、登録確認プログラムGP08(u)、年金管理プログラムGP09、維持要否判断プログラムGP10(u)、権利満了処理プログラムGP11、放棄処理プログラムGP12(u)と構成されている。この例では、例えば公開確認プログラムGP01はベンダ提供パッケージのままであるが、審査請求判断プログラムGP02(u)はベンダ提供パッケージをカスタマイズしたプログラムである。

【0019】この様子をさらに具体的に、図4に例として審査請求判断プログラムGP02(u)の構成例によって説明する。図4(a)には二つのユーザa, bの審査請求判断仕様を示す。すなわち、ユーザaでは社内審査分類として当該発明を(重点度a, b, c)に分類し、(補正要、否:請求要、不要)の決定をする。そして、重点a

は即請求、重点bは一次時点で請求、重点cは二次時点で請求することが社内ルールである。ここに一次時点とはこのユーザの場合出願後3年時点、二次時点とは出願後6年時点と決められている。一方、ユーザbの審査請求判断仕様では、社内審査分類として当該発明を(補正要、否)および(即時審査請求、1年前見直し時点まで保留、請求不要)の決定をする。

【0020】これらの仕様を反映した業務プログラムGP02(a), GP02(b)の構成例は図4(b)のようになる。ここにルーチンK1, K2, K3, K4はベンダの提供するGP02プログラムのコア部を形成するもので、図3(c)にその内容が示される。すなわち、ルーチンK1は特許管理クライアント1'から該当発明の整理番号、社内審査分類を受け入れ、該当発明の出願日、現日付を求める。ルーチンK2は一次時点の日付、二次時点の日付、審査請求期限日付、一次時点残日数、二次時点残日数、期限日残日数、期限日1年前日、期限日1年前残日数などを計算する。ここでは、まず出願日から適用法令を探し、該当法令の定める審査請求期限日を求め、これをもとに社内ルールで定めた一次時点の日付、二次時点などの日付を計算する。ルーチンK2設計にあたっては、一次時点、二次時点としていかなる時点をとるかを全ユーザの希望を満たすように種々与えることができるよう工夫する。

【0021】図4(b)のユーザaの業務プログラムGP02(a)ではルーチンK1, K2によって期限に関するチェックを実行した後、仕分けルーチンS(a)で当該発明を(重点a, b, c, 請求不要、期限エラー)に仕分ける。この部分はユーザaの社内仕様にカスタマイズされたルーチンである。次いで、GP02プログラムのコア部のルーチンK3(期限エラーの場合はリトライを促し、確認クリックの場合は次に進む)、ルーチンK4(特許データベース更新手段23'をコールしデータベース更新を行う)を実行して終了する。

【0022】一方、図4(b)のユーザbの業務プログラムGP02(b)でもルーチンK1, K2, K3, K4の部分はユーザaと同様であり、仕分けルーチンS(b)のみが仕分けルーチンS(a)と異なり、当該発明を(請求即、1年前見直し、不要、期限エラー)に仕分ける。この部分はユーザbの社内仕様にカスタマイズされたルーチンである。図3に戻って、図3(a)のような従来のシステムにおいて、法令改訂があった場合のバージョン更新に伴う必要作業を図3(b)に示す。必要作業はバージョン更新作業v-wkと各ユーザ向けカスタマイズ作業c-wkとである。バージョン更新作業v-wkとしては提供パッケージの全業務にわたり法令ロジックチェックが関わっているため全コア部の見直しが必要である。さらにこれをユーザに適用する場合、該当ユーザのカスタマイズプログラム(図3(a)の例では審査請求判断プログラムGP02(u)、中間処理プログラムGP05(u)、登録確認プログラムGP08(u)、維持要否判断プログラムGP10(u)、放棄処理プログラムGP12

(u) ) において各ユーザ向けカスタマイズ作業c-wkが必要である。

【0023】本発明はこのような法令改訂に伴う作業範囲の減少を図り、もって作業工数の削減とミスによる改訂反映漏れを減少を期するものである。図4で示した審査請求判断プログラムGP02(u)の従来例を題材にして、本発明における実施例を図5に示す。本発明ではまず、図5(a)に示す如く、従来の業務プログラムを見直し、内在するビジネスロジック部分と法令ロジック部分を分離したプログラムスタイルをとる。具体的には、図4の例ではルーチンK1, K3は法令ロジックを含まないのでビジネスロジック部であり、ルーチンK4は法令ロジック部であるが、ルーチンK2および仕分けルーチンS(a), S(b)において法令ロジック部分が混在しているので、図のように分離する。

【0024】すなわち、ルーチンK2から法令ロジック部分である期限日を計算するルーチンをルーチンK2Lとして分離し、残余をビジネスロジックルーチンK2Bとする。図式的には $K2=K2B+K2L$ と表される。同様に、仕分けルーチンS=仕分けビジネスロジックルーチンSB+法令ロジックルーチンSLと分離する。すなわち、図式的には $S(a)=SB(a)+SL$ 、 $S(b)=SB(b)+SL$ である。ここにSLは仕分けに関するユーザ共通の法令ロジックチェック部分であり、法令に依存するがユーザ仕様には依存しない部分である。

【0025】このように、ビジネスロジックチェック部分と法令ロジックチェック部分とを分離してプログラムを作成することにより、法令改訂時のバージョン更新範囲の減少が図られる。これを図6で説明する。すなわち、図6(a)のように、本発明の特許管理システムはビジネスロジックチェック手段21(プログラムBC(u))と法令ロジックチェック手段22(プログラムLC(u))とを分離して持つ。このような構成にした場合、法令改訂時のバージョン更新範囲は図6(b)に示すようになる。これを従来例の図3(b)と比較すると、一般にLCコア部サイズ $\langle GP01+GP02+\dots+GP12 \rangle$ コア部サイズが成立するので、ベンダにおけるバージョン更新作業v-wkの範囲の減少が図られ、作業負荷の減少が図られている。またカスタマイズ部についても、一般にLC(u)サイズ $\langle GP01(u)+GP02(u)+\dots+GP12(u) \rangle$ サイズであるから、 $(LC(a)+\dots+LC(e))$ サイズ $\langle (GP01(a)+GP02(a)+\dots+GP12(a))+\dots+(GP01(e)+GP02(e)+\dots+GP12(e)) \rangle$ サイズである。従って、法令改訂に伴う各ユーザ向けカスタマイズ作業c-wk負荷の減少も図られている。

【0026】再び図5に帰って、本発明の各ユーザ向けカスタマイズの構造を図5(b)で説明する。ユーザa向け業務プログラムGP02(a)を実質的に動作させるプログラムは図4および図5(a)から、K1, K2B, K2L, SB(a), SL, K3, K4であることがわかる。このうち、K1, K2B, SB(a), K3がビジネスロジックチェック手段21に含まれ、K2L, S

L, K4は法令ロジックチェック手段22に含まれることは既に述べた通りである。ビジネスロジックチェック手段21中のK1, K2B, SB(a), K3に着目すると、K1, K2B, K3はコア部でありSB(a)のみがユーザa向けカスタマイズ部である。本発明ではこれらルーチンをオブジェクト指向技術によりオブジェクト集合としてインプリメントする。そして、コアオブジェクトをカスタマイズすることをオブジェクト間の継承関係を利用して実現する。

【0027】図5(b)によってその仕組みを説明する。本発明では図5(b)のようにビジネスロジックチェック手段21をビジネスロジックチェックカスタマイズ部BC-Cとビジネスロジックチェック共通部BC-Kとに分離する。また、法令ロジックチェック手段22についても、法令ロジックチェックカスタマイズ部LC-Cと法令ロジックチェック共通部LC-Kとに分離する。このとき、ベンダで提供する実効メソッドを内蔵するオブジェクトを抽象クラスとし、これを共通部(例えばビジネスロジックチェック共通部BC-K, 法令ロジックチェック共通部LC-K)とし、これを実行させるインタフェースとしての実行クラスをカスタマイズ部(例えばビジネスロジックチェックカスタマイズ部BC-C, 法令ロジックチェックカスタマイズ部LC-C)として提供する。このように提供された実行クラスのオブジェクトは、自らはメソッドを内蔵せず、起動メッセージを受けた場合、親クラスである抽象クラスをたどり、該当のオブジェクトのメソッドを実行する。そして、カスタマイズが必要な場合はこのカスタマイズ部の実行クラスオブジェクトをカスタマイズ内容をメソッドとして記述したカスタマイズオブジェクトに置き換える。

【0028】以上の関係をユーザa向け業務プログラムGP02(a)の場合で説明すると、図5(b)のように、ビジネスロジックチェック共通部BC-KにはK1, K2B, K3, SB、法令ロジックチェック共通部LC-KにはK2L, SL, K4の抽象クラスオブジェクトが用意されている。そしてこれらに対応する実行クラスオブジェクトK1', K2B', K3', SB', K2L', SL', K4'がカスタマイズ部に準備されている。これに対して、ユーザaは共通部の仕分けルーチンSB+SLをユーザ仕様によってSB(a)+SLに変更したい場合、抽象クラスのSBを変更するのではなく、実行クラスのSB'をSB(a)に置き換えるのである。

【0029】これによってユーザaとユーザbのシステムでは共通のビジネスロジックチェック共通部BC-Kおよび法令ロジックチェック共通部LC-Kを有し、ベンダは全ユーザに唯一つの共通のビジネスロジックチェック共通部BC-K, 法令ロジックチェック共通部LC-Kをサポートすればよい。かつ、ビジネスロジックチェック共通部BC-Kは法令改訂の影響を受けない構造となっているので、法令改訂にあたっては、ベンダは唯一つの法令ロジックチェック共通部LC-Kに対してバージョン更新作業v-wkを行えばよい。また、この構造では法令改訂の内容はオブジ



エクトの継承関係によってユーザカスタマイズ部に自動的に反映する。

【0030】以上の説明を図7によって復唱すると、図7(a)の実施例のシステムでは特許管理サーバ2のビジネスロジックチェック手段21はビジネスロジックチェックカスタマイズ部BC-Cとビジネスロジックチェック共通部BC-Kとに分離した構造であり、法令ロジックチェック手段22は法令ロジックチェックカスタマイズ部LC-Cと法令ロジックチェック共通部LC-Kとに分離している。この状態における法令改訂バージョン更新作業の範囲は図7(b)に示すようにフロントエンドチェック共通部FC-K、ビジネスロジックチェック共通部BC-K、法令ロジックチェック共通部LC-Kに限定されるので、従来よりもバージョン更新作業v-wkの範囲の減少が図られ、作業負荷の減少が図られている。さらに、FC-C(u)、BC-C(u)、LC-C(u)はFC-K、BC-K、LC-Kのバージョン更新作業v-wkの内容を継承する実行クラスなので法令改訂に伴う変更は不要、従って、ユーザカスタマイズ工数が不要である。

【0031】さらに、第3の実施例について図8により説明する。図8(a)の特許管理システムは法令ロジックチェック手段22の中で、さらに法令データベースL-DBを分離した構造である。法令データベースL-DBは法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成したものであり、図10にその構造例を示す。

【0032】図10の実施例では図10(a)に示すように法令データベースL-DBは個別の法令に対応したデータベースファイル(日本特許法.db、米国特許法.db、...)の集合である。例えば、日本特許法に対応する法令データベースL-DBの構造は図10(b)に示すように、過去のすべての改訂を含むものである。図は平成6年改正時点の例である。図のように法令ロジックチェックに必要な項目が大項目、中項目、小項目に整理され、各項目は必要に応じて複数列作られる。例えば審査請求期限については、平成5年改正以前から『出願日から7年』であり、平成6年改正にあっても不変であるので、この欄は『ap---07y00m00d』と記入された1セルだけでよい。しかし、平成11年改正によって『出願日から3年』となるので、平成11年改正を反映すると『出願日<2001年10月1日なら出願日から7年』、『出願日>2001年10月1日なら出願日から3年』の2つを選択して適用しなければならない。このことを法令データベースL-DB上に反映すると、『適用ルール: ap, 011001、期限: ap---07y00m00d』および『適用ルール: 011001, ap、期限: ap---03y00m00d』の2項目2列に書き換えられることとなる。

【0033】図8に戻ると、図8(a)のように構成された特許管理システムでは、図8(b)に示すように、法令改訂にあたって法令ロジックチェック共通部LC-Kすらもバージョン更新作業の対象ではなく、法令データベース

L-DBのみをバージョン更新作業v-wkの範囲とすることができる。これは法令の枠組みが変わらない範囲の改訂であって、例えば、審査請求制度そのものがなくなったり新たな手続が挿入されるような大改訂の場合には法令ロジックチェック共通部LC-Kにもバージョン更新作業が必要となることが想定されるが、現在までの枠組み変化の実績に対しては法令データベースL-DBの変更で対処できている。

【0034】次に、法令改訂に伴う法令データベースL-DBの変更作業について検討する。本発明では図1に示すような法令データベースL-DBを更新する法令データベース更新サーバ3を提案する。次に図12を併用して法令データベース更新サーバ3の動作例を説明する。改訂法令登録更新時処理について図12(a)の動作フローにより説明する。法令データベース更新サーバ3は改訂法令登録手段32を有しており、ステップS1201によって改訂法令登録手段32は改訂法令データベース情報L-DB'を読み込み法令データベース更新サーバ3内に格納する。改訂法令データベース情報L-DB'は図10(b)の構造に適するように変更するデータ部分のみを書き換える情報としてベンダエンジニアによって作成され改訂法令登録手段32に投入されたものである。次いで法令データベース更新サーバ3の図示しないメール手段はオペレータ指示によりステップS1202においてサポートユーザリストから今回改訂の法令を使用しているユーザを抽出する。このサポートユーザリストはベンダからユーザに特許管理システムが納入される毎に、そのカスタマイズ情報やユーザ責任者メールアドレス、ユーザ担当者メールアドレスなどとともに法令データベース更新サーバ3に記録更新されるものである。次いで法令データベース更新サーバ3の図示しないメール手段はステップS1202で決定したユーザをステップS1203で送信対象ユーザとして未受信ユーザリストを生成し印刷すると共に、ステップS1204でこれらユーザに更新案内メールを作成しユーザ担当者メールアドレスに自動送付する。更新案内の内容は、例えば改訂概要、適用開始日付、ダウンロード方法、インストール方法などを知らせるものである。

【0035】上記メールを受信したユーザ担当者が法令データベース更新サーバ3のホームページにダウンロード要求を送ることによって、図12(c)のダウンロード要求受信時処理が開始する。法令データベース更新サーバ3の改訂法令データベース情報送信手段31はステップS1209によってダウンロード要求ユーザのチェックを行い、認証処理によって正しいユーザであること、ダウンロード費用の契約関係が処理済であることを確認する。その結果をステップS1210で判定し、送信許可と判定されたならば、改訂法令データベース情報送信手段31はステップS1211で当該改訂法令データベース情報L-DB'を送信する。そして送信が成功したならば未受信ユーザリストに送信済を記録して消し込みを行い、未受信ユーザ

リストを更新する。

【0036】さらに、法令データベース更新サーバ3の図示しない共通制御部は、オペレータ指定期間中は毎日定時に図1.2(b)に示す日次処理を起動する。すなわち、法令データベース更新サーバ3のアラーム先判定手段33はステップS1205で現在時点の未受信ユーザリストを表示し、ステップS1206で当該改訂法令適用日までの残日数を計算し、残日数が30日である場合にはステップS1207を、残日数が15日であるならばステップS1208を起動する。

【0037】ステップS1207では法令データベース更新サーバ3のアラーム通知手段34は当該未受信ユーザの責任者および担当者にアラームメールを自動送付する。その内容としては当該法令改訂を適用しない場合の影響範囲や責任事項、改訂法令データベース情報-L-DB'インストール、テスト所要期間予想などを示し、回答の返信を依頼する。なお、図1.2に図示しないが、上記回答によって未受信でよいことが確認された場合は法令データベース更新サーバ3のオペレータはマニュアルで当該ユーザを未受信ユーザリストから抹消する。ステップS1208ではアラーム通知手段34は未受信ユーザをサポートするベンダの担当営業責任者、営業担当者にコンタクト依頼メールを自動送付する。メール内容は当該ユーザにコンタクトして早急な対応をとるように依頼するものである。

【0038】以上説明した法令データベース更新サーバ3に対して、図9に示す本発明の特許管理システムの実施例では特許管理サーバ2は法令データベース受信更新手段24を有して、前記法令データベース更新サーバ3からの改訂法令データベース情報-L-DB'を受信し法令データベースL-DBに上書きする。このような実施例では改訂法令データベース情報-L-DB'によって法令データベースL-DBを更新するバージョン更新作業v-wkがユーザサイトで実行可能となる。従ってユーザは従来のように更新パッケージインストール作業経費をベンダに支払う必要がない。

【0039】改訂法令データベース情報-L-DB'のデータ形式については任意であるが、本発明の実施例として図11に示すようなXML形式による改訂法令データベース情報-L-DB'の表現例を説明する。本形式は図10(b)のデータ構造に対応するもので、大項目、中項目、小項目のそれぞれの項目名をタグとしデータ値が開始/終了タグ間に記載される。

【0040】図11(a)は平成6年改正時の改訂法令データベース情報-L-DB'の抜粋例であり、本改訂法令データベース情報-L-DB'が格納された状態が図10(b)の満了ルール部分に対応する。また、先に説明した平成11年改正の図10(b)の※印に対応する改訂法令データベース情報-L-DB'の抜粋が図11(b)に示されるものである。データ中のap, pb, rg, or はそれぞれ出願日、公告

日、登録日、基準出願日を示す変数名であり、例えば図11(a)の三つの満了ルールA, B, Cはそれぞれ下記のようなルールを表現している。

満了ルールA：本ルールは特許データベースP-DB中に公告日を入力するアクションで起動し、出願日<1994年1月1日であるならば公告日から15年を権利満了と設定せよ。

満了ルールB：本ルールは特許データベースP-DB中に公告日を入力するアクションで起動し、1994年1月1日<出願日<1995年7月1日であるならば、出願日から20年、公告日から15年のうち短い方の期間を権利満了と設定せよ。

満了ルールC：本ルールは特許データベースP-DB中に登録日を入力するアクションで起動し、1995年7月1日<出願日であるならば基準出願日から20年を権利満了と設定せよ。

【0041】このような形式を採用することにより書き換え必要項目のみを分かりやすいフォームで送信することができる。また、このようなオープン形式データは本発明の特許管理システム以外でもデータとして応用可能であり、一般に有料コンテンツとしての流通可能性を持つものである。

(付記1) 法令依存物件の帰趨履歴を格納する法令依存物件データベースを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバと、前記法令依存物件データベースへのデータ登録、更新入力を行う1つ以上の法令依存物件管理クライアントとから構成される法令依存物件管理システムにおいて、法令依存物件管理サーバのコンピュータに法令依存物件管理情報処理を実行させるための法令依存物件管理プログラムであって、前記法令依存物件管理クライアントからの前記法令依存物件データベースへのデータ登録、更新のためのデータ入力に対して、管理業務における予め定められたルールであるビジネスロジックに従ってチェックを行い、エラーがあれば再入力を促し、ビジネスロジックとしてのエラーのないデータのみを受け入れるビジネスロジックチェック手段としてコンピュータを機能させるビジネスロジックチェックプログラムと、前記データ入力に対して、法令における予め定められたルールである法令ロジックに照らしてエラーがあれば再入力を促し、法令ロジックエラーのないデータのみを受け入れる法令ロジックチェック手段としてコンピュータを機能させる法令ロジックチェックプログラムと、を有し、前記ビジネスロジックチェックプログラムと前記法令ロジックチェックプログラムとが分離していることを特徴とする法令依存物件管理プログラム。

【0042】(付記2) 前記ビジネスロジックチェックプログラムは、さらに、ビジネスロジックチェック処理を共通的に記述する抽象クラスオブジェクト集合であるビジネスロジックチェック共通部と、前記ビジネスロ

ジックチェック共通部を実行するための実行クラスからなりユーザ毎のシステム環境や業務条件に適合するようにカスタマイズされたビジネスロジックチェックカスタマイズ部と、に分離されたプログラムであり、前記法令ロジックチェックプログラムは、さらに、法令ロジックチェック処理を共通的に記述する抽象クラスオブジェクト集合である法令ロジックチェック共通部と、前記法令ロジックチェック共通部を実行するための実行クラスからなりユーザ毎のシステム環境や業務条件に適合するようにカスタマイズされた法令ロジックチェックカスタマイズ部と、に分離されたプログラムであることを特徴とする付記 1 記載の法令依存物件管理プログラム。

【0043】（付記 3） 前記法令ロジックチェック共通部は、さらに、法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した法令データベースを外部に有し、当該法令データベースを参照して法令ロジックチェックを行う構成であることを特徴とする付記 2 記載の法令依存物件管理プログラム。

【0044】（付記 4） 法令依存物件の帰趨履歴を格納する法令依存物件データベースを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバと、前記法令依存物件データベースへのデータ登録、更新入力を行う 1 つ以上の法令依存物件管理クライアントとから構成される法令依存物件管理システムであって、前記法令依存物件データベースへのデータ登録、更新のためのデータ入力のフォーマットチェックを行い、エラーがあれば再入力を促し、フォーマットエラーのないデータのみを受け入れるフロントエンドチェック手段と、付記 1 または 2 または 3 記載の法令依存物件管理プログラムによって機能するビジネスロジックチェック手段および法令ロジックチェック手段を有する法令依存物件管理サーバと、を有することを特徴とする法令依存物件管理システム。

【0045】（付記 5） 改訂された法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した改訂法令データベース情報として登録する改訂法令登録手段と、法令依存物件管理システムに対してインターネットを通じて前記改訂法令データベース情報を送信する改訂法令データベース情報送信手段とを有する法令データベース更新サーバ。

【0046】（付記 6） 付記 5 記載の法令データベース更新サーバであって、さらに、前記法令依存物件管理システムのカスタマイズ情報と前記改訂法令データベース情報と現在日時および送信済履歴とを対比して、前記改訂法令データベース情報の受信が必要であるのに未受信である法令依存物件管理システムを抽出するアラーム先判定手段と、前記アラーム先判定手段が抽出した法令依存物件管理システムに対してアラーム通知を行うアラーム通知手段と、を有することを特徴とする法令データ

ベース更新サーバ。

【0047】（付記 7） 付記 5 または 6 記載の法令データベース更新サーバからインターネットを通じて改訂法令データベース情報を受信し法令データベースを更新する法令データベース受信更新手段を有することを特徴とする法令依存物件管理システム。

（付記 8） 法令依存物件の帰趨履歴を格納する法令依存物件データベースを有し、法令に基づいて必要となる手続を管理する法令依存物件管理サーバと、前記法令依存物件データベースへのデータ登録、更新入力を行う 1 つ以上の法令依存物件管理クライアントとから構成される法令依存物件管理システムの法令依存物件管理方法であって、前記法令依存物件データベースへのデータ登録、更新のためのデータ入力のフォーマットチェックを行い、エラーがあれば再入力を促し、フォーマットエラーのないデータのみを受け入れるフロントエンドチェック段階と、前記法令依存物件管理クライアントからの前記法令依存物件データベースへのデータ登録、更新のためのデータ入力に対して、管理業務における予め定められたルールであるビジネスロジックに従ってチェックを行い、エラーがあれば再入力を促し、ビジネスロジックとしてのエラーのないデータのみを受け入れるビジネスロジックチェック段階と、前記データ入力に対し法令における予め定められたルールである法令ロジックに照らしてエラーがあれば再入力を促し、法令ロジックエラーのないデータのみを受け入れる法令ロジックチェック段階と、を有することを特徴とする法令依存物件管理方法。

【0048】（付記 9） 付記 1 または 2 または 3 記載の法令依存物件管理プログラムを記録したコンピュータ読み取り可能な記録媒体。

（付記 10） 法令依存物件管理システムに対して改訂法令データベース情報を供給する情報処理を法令データベース更新サーバに実行させる法令データベース更新サーバプログラムであって、改訂された法令の所定の記述をコンピュータにロジックチェックさせるために適した項目の集合として構成した改訂法令データベース情報として登録する改訂法令登録段階と、法令依存物件管理システムに対してインターネットを通じて前記改訂法令データベース情報を送信する改訂法令データベース情報送信段階と、を法令データベース更新サーバに実行させることを特徴とする法令データベース更新サーバプログラム。

【0049】（付記 11） 付記 10 記載の法令データベース更新サーバプログラムであって、さらに、前記法令依存物件管理システムのカスタマイズ情報と前記改訂法令データベース情報と現在日時および送信済履歴とを対比して、前記改訂法令データベース情報の受信が必要であるのに未受信である法令依存物件管理システムを抽出するアラーム先判定段階と、前記アラーム先判定手段

10

20

30

40

50

が抽出した法令依存物件管理システムに対してアラーム通知を行うアラーム通知段階と、を法令データベース更新サーバに実行させることを特徴とする法令データベース更新サーバプログラム。

【0050】（付記12） 法令依存物件管理システムコンピュータに法令依存物件管理を実行させるための法令依存物件管理プログラムであって、付記5または6記載の法令データベース更新サーバからインターネットを通じて改訂法令データベース情報を受信し法令データベースを更新する法令データベース受信更新段階をコンピュータに実行させることを特徴とする法令依存物件管理プログラム。

【0051】

【発明の効果】以上の説明から明らかなように、本発明によれば、法令改訂に伴うパッケージのバージョンアップ範囲をより減少させ、作業工数の削減とミスによる改訂反映漏れを減少させる特許管理システムの提供が可能となる、という効果がある。また、パッケージのバージョンアップに伴うユーザカスタマイズの作業工数を削減させる特許管理システムの提供が可能となる、という効果がある。また、本発明の実施例では特許管理システムにおける詳細な内容を説明したが、特許管理システムだけでなく、一般に、法令に基づいて必要となる手続を行う法令依存物件を管理する法令依存物件管理システム全般にいえることである。

【図面の簡単な説明】

【図1】本発明の法令依存物件管理システムおよび法令データベース更新サーバの構成例

【図2】特許データベースP-DBを更新する特許管理業務一覧

【図3】従来の特許管理システムとバージョン更新時の作業範囲例

【図4】従来の業務プログラム（GP02(u) 審査請求判断）の構成例

【図5】本発明実施例における（GP02(u) 審査請求判断）部分の構成例

【図6】特許管理システムの第1の実施例でのバージョ

ン更新時の作業範囲例

【図7】特許管理システムの第2の実施例でのバージョン更新時の作業範囲例

【図8】特許管理システムの第3の実施例でのバージョン更新時の作業範囲例

【図9】特許管理システムの第4の実施例でのバージョン更新時の作業範囲例

【図10】法令データベースL-DBの構造例

10 DB' の表現例

【図12】法令データベース更新サーバの動作例フロー図

【符号の説明】

1 法令依存物件管理クライアント（特許管理クライアント）

2 法令依存物件管理サーバ（特許管理サーバ）

3 法令データベース更新サーバ

11 フロントエンドチェック手段

21 ビジネスロジックチェック手段

20 22 法令ロジックチェック手段

23 法令依存物件データベース更新手段（特許データベース更新手段）

24 法令データベース受信更新手段

31 改訂法令データベース情報送信手段

32 改訂法令登録手段

33 アラーム先判定手段

34 アラーム通知手段

FC-K フロントエンドチェック共通部

FC-C フロントエンドチェックカスタマイズ部

30 BC-K ビジネスロジックチェック共通部

BC-C ビジネスロジックチェックカスタマイズ部

LC-K 法令ロジックチェック共通部

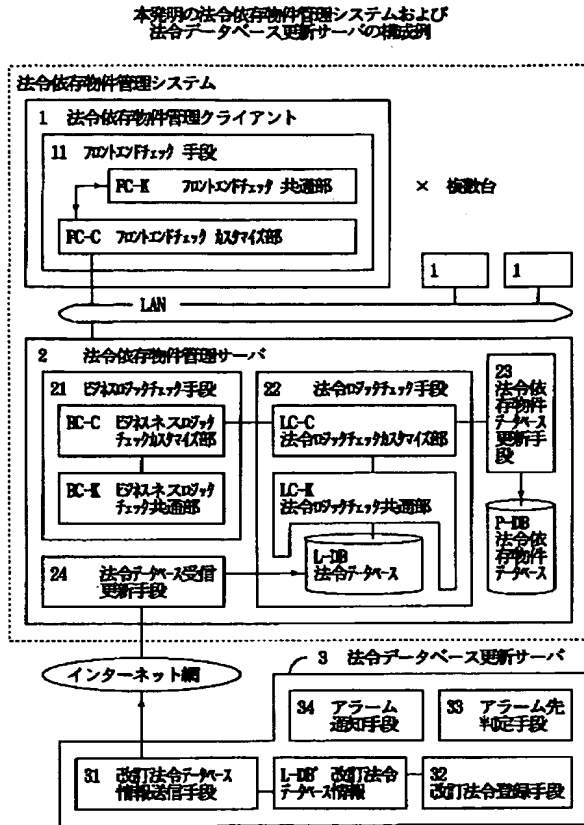
LC-C 法令ロジックチェックカスタマイズ部

L-DB 法令データベース

L-DB' 改訂法令データベース情報

P-DB 法令依存物件データベース（特許データベース）

【図 1】



【図 2】

特許データベースPDBを更新する特許管理業務一覧

- (a) 特許管理業務のシステム化の全貌
1. 特許管理システム (①-④ 公報の蓄積、検索機能)
  2. 特許管理システム (⑤-⑧ 特許の蓄積、検索機能、特許の蓄積)
  3. 特許管理システム (⑨-⑫ 特許の蓄積、検索機能、特許の蓄積)
  4. 特許管理システム (⑬-⑯ 特許の蓄積、検索機能、特許の蓄積)

(b) 特許管理システムの特許データベースPDBを更新する特許管理業務一覧

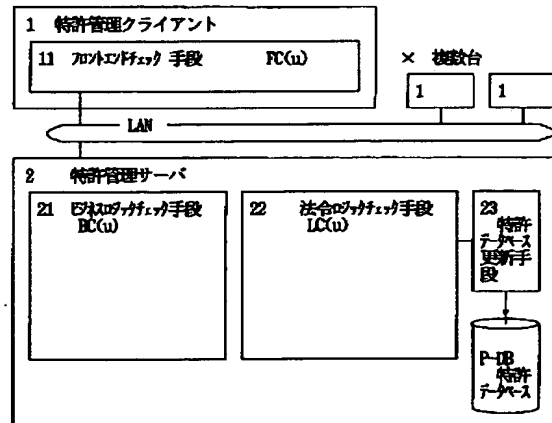
※はユーザカスタマイズ部分が多い業務と見られるもの。

| 業務名      | PDB<br>による<br>更新 | 法令700ポートによる更新 |    |    |    |    |    |    |
|----------|------------------|---------------|----|----|----|----|----|----|
|          |                  | 日特            | 日実 | 米特 | 米実 | 欧特 | 欧実 | 欧特 |
| ※ 提案受付   | ○                |               |    |    |    |    |    |    |
| ※ 前次更新手段 | ○                |               |    |    |    |    |    |    |
| ※ 事務所依頼  | ○                |               |    |    |    |    |    |    |
| ※ 出願手段   | ○                |               |    |    |    |    |    |    |
| ※ 公報蓄積   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 審査請求手続 | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 審査請求   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 庁通知処理  | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 中絶処理   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 補正手続   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 審判手続   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 登録蓄積   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 年金管理   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 維持費手続  | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 権利満了処理 | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |
| ※ 放棄処理   | ○                | ○             | ○  | ○  | ○  | ○  | ○  | ○  |

【図 6】

特許管理システムの第1の実施例でのバージョン更新時の作業処理例

(a) 第1の実施例のユーザ用カスタマイズされた特許管理システム



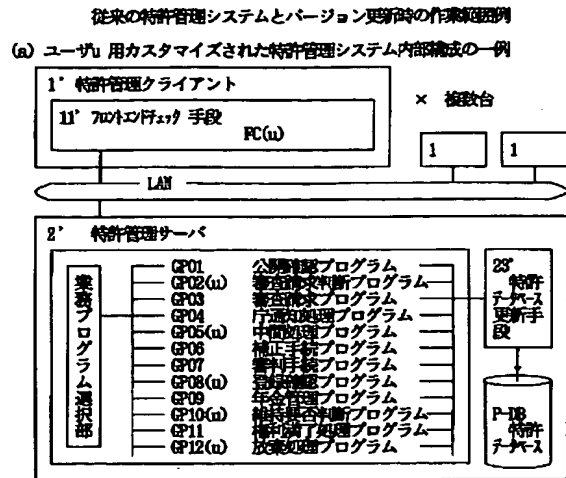
(b) 第1の実施例の法令改訂に伴うバージョン更新時の作業処理例

| バージョン更新作業<br>ユーザ | 各ユーザ向けカスタマイズ作業ユーザ | ユーザa           | ユーザb           | ユーザc           | ユーザd           | ユーザe           |
|------------------|-------------------|----------------|----------------|----------------|----------------|----------------|
| FCコア部<br>LCコア部   | FC(a)<br>LC(a)    | FC(b)<br>LC(b) | FC(c)<br>LC(c) | FC(d)<br>LC(d) | FC(e)<br>LC(e) | FC(e)<br>LC(e) |

一般に LCコア部サイズ < (GP1+GP2+...+GP12) コア部サイズ  
ゆえに バージョン更新作業一貫負荷の減少が図られる。

一般に (LC(a)+...+LC(e)) サイズ < ((GP1(a)+GP2(a))+...+GP12(a))+...+GP12(e)) サイズ  
ゆえに 各ユーザ向けカスタマイズ作業一貫負荷の減少が図られる。

【図 3】



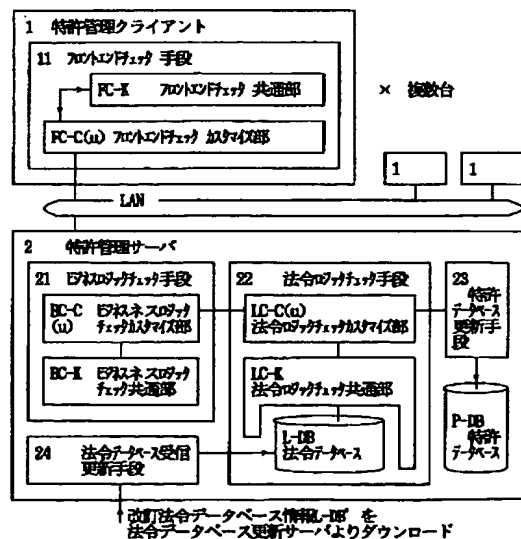
(b) 従来システムでの法令改訂に伴うバージョン更新時の作業手順

| バージョン更新作業<br>v-ek | 各ユーザ向けカスタマイズ作業-ek | ユーザa<br>FC(a) | ユーザb<br>FC(b) | ユーザc<br>FC(c) | ユーザd<br>FC(d) | ユーザe<br>FC(e) |
|-------------------|-------------------|---------------|---------------|---------------|---------------|---------------|
| FCコア部             | FC(a)             | FC(b)         | FC(c)         | FC(d)         | FC(e)         | FC(e)         |
| GP01コア部           | GP02(a)           | GP02(b)       | GP02(c)       | GP02(d)       | GP02(e)       | GP02(e)       |
| GP02コア部           | GP05(a)           | GP05(b)       | GP05(c)       | GP05(d)       | GP05(e)       | GP05(e)       |
| GP03コア部           | GP08(a)           | GP08(b)       | GP08(c)       | GP08(d)       | GP08(e)       | GP08(e)       |
| GP04コア部           | GP10(a)           | GP10(b)       | GP10(c)       | GP10(d)       | GP10(e)       | GP10(e)       |
| GP05コア部           | GP12(a)           | GP12(b)       | GP12(c)       | GP12(d)       | GP12(e)       | GP12(e)       |
| GP06コア部           |                   |               |               |               |               |               |
| GP07コア部           |                   |               |               |               |               |               |
| GP08コア部           |                   |               |               |               |               |               |
| GP09コア部           |                   |               |               |               |               |               |
| GP10コア部           |                   |               |               |               |               |               |
| GP11コア部           |                   |               |               |               |               |               |
| GP12コア部           |                   |               |               |               |               |               |

【図 9】

特許管理システムの第4の実施例でのバージョン更新時の作業手順例

(a) 第4の実施例のユーザ用カスタマイズされた特許管理システム

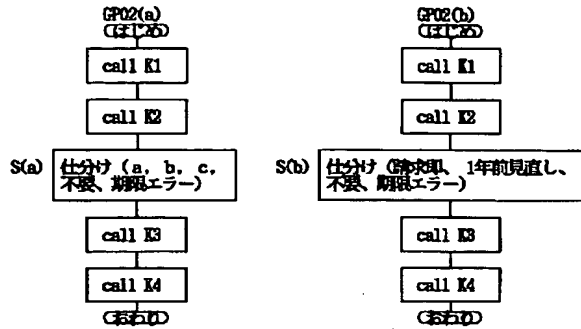
(b) 第4の実施例の法令改訂に伴うバージョン更新時の作業手順  
改訂法令データベース情報-DB によって法令データベース-DBを更新するバージョン更新作業-ekがユーザサイトでも実行可能となる。従ってユーザは従来のように更新パッケージインストール作業をベンダーに支払う必要がない。

【図 4】

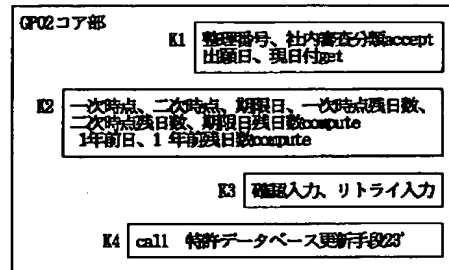
従来の業務プログラム (GP02(w) 審査請求手続) の構成例

(a) ユーザ仕様例  
 ユーザa : 社内審査分類 (重点度 a, b, c : 補正要、否 : 請求要、不要)  
 重点a : 即決要、重点b : 一次時点請求、重点c : 二次時点請求  
 ユーザb : 社内審査分類 (補正要、否 : 請求要、1年前見直し、不要)

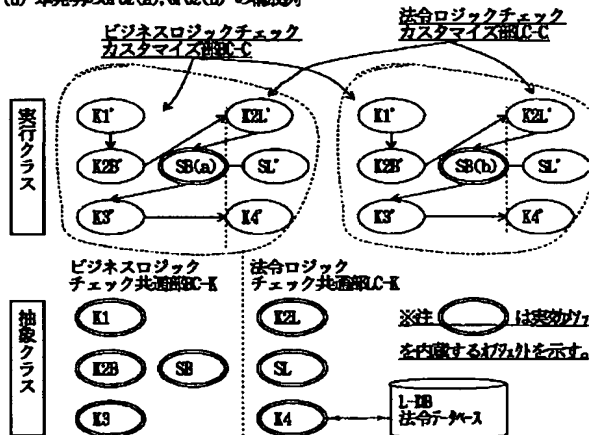
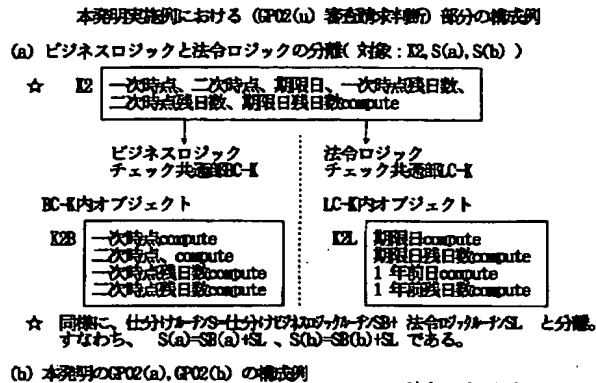
(b) GP02(a), GP02(b) の構成例



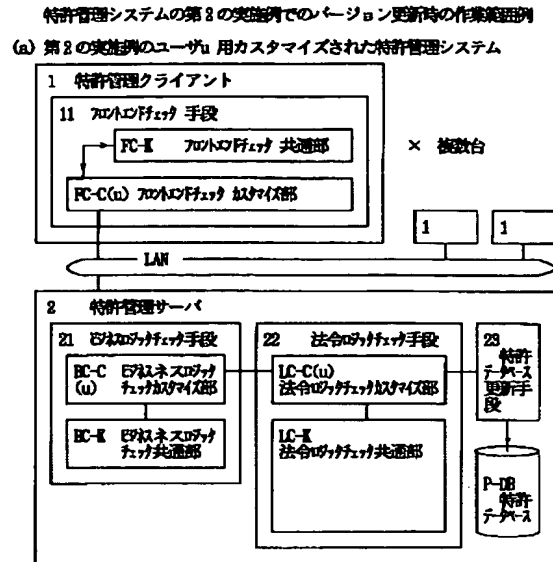
(c) GP02 コア部の構成例



【図 5】



【図 7】



(b) 第 2 の実施例の法令改訂に伴うバージョン更新時の作業範囲

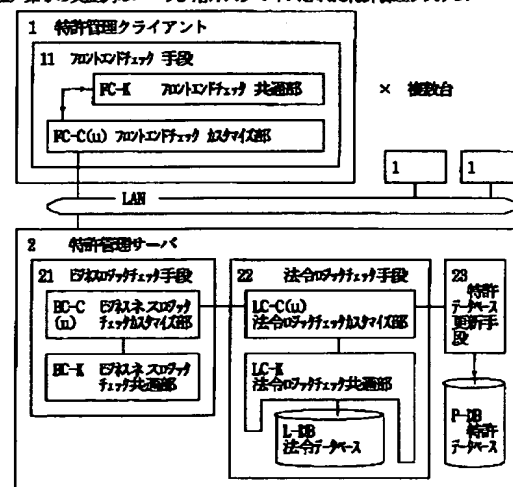
| バージョン更新作業<br>v-uk | 各ユーザ向けカスタマイズ作業-uk |       |       |       |       |
|-------------------|-------------------|-------|-------|-------|-------|
|                   | ユーザ a             | ユーザ b | ユーザ c | ユーザ d | ユーザ e |
| FC-I              | 変更不要              | 変更不要  | 変更不要  | 変更不要  | 変更不要  |
| LC-I              | 変更不要              | 変更不要  | 変更不要  | 変更不要  | 変更不要  |

FC-C(u), LC-C(u) は抽象クラスである FC-I, LC-I のバージョン更新作業-uk の内容を継承する実行クラスなので変更不要である。

【図 8】

特許管理システムの第 3 の実施例でのバージョン更新時の作業範囲

(a) 第 3 の実施例のユーザ用カスタマイズされた特許管理システム



(b) 第 3 の実施例の法令改訂に伴うバージョン更新時の作業範囲

| バージョン更新作業<br>v-uk | 各ユーザ向けカスタマイズ作業-uk |       |       |       |       |
|-------------------|-------------------|-------|-------|-------|-------|
|                   | ユーザ a             | ユーザ b | ユーザ c | ユーザ d | ユーザ e |
| L-DB              | 変更不要              | 変更不要  | 変更不要  | 変更不要  | 変更不要  |

法令の特組がかわらない改訂の場合はバージョン更新作業-uk は法令データベース L-DB の更新のみで法令ロジックチェック共通部 LC-I は不要でよい。その場合、バージョン更新作業-uk は第 2 の実施例に比して一層小さな範囲に絞られる。





## フロントページの続き

(72) 発明者 伊藤 邦章  
長野県長野市大字鶴賀字鍋屋田1403番地 3  
株式会社富士通長野システムエンジニア  
リング内  
(72) 発明者 小川 彰人  
長野県長野市大字鶴賀字鍋屋田1403番地 3  
株式会社富士通長野システムエンジニア  
リング内

(72) 発明者 竹原 健  
神奈川県川崎市中原区上小田中 4 丁目 1 番  
1 号 富士通株式会社内  
(72) 発明者 阿部 豊  
長野県長野市大字鶴賀字鍋屋田1403番地 3  
株式会社富士通長野システムエンジニア  
リング内

